

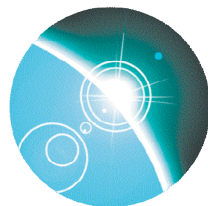
# The Future of NetCDF

Russ Rew

UCAR Unidata Program Center

Acknowledgments: John Caron, Ed Hartnett,  
NASA's Earth Science Technology Office,  
National Science Foundation

GO-ESSP Meeting, June 2005



unidata

# Overview

- What is netCDF?
- What is netCDF-4?
- What's new in the data model?
- How are the APIs changing?
- What new capabilities will be available?
- Are there implications for conventions?

# What is NetCDF?

- A **Data Model** for scientific data: variables, dimensions, attributes, coordinates
- **Application Programming Interfaces** for data access in C, Fortran, Java, C++, Perl, Python, Ruby, ...
- A **Format** for self-describing portable binary data

*Users need not know anything about the format*

# NetCDF Principles

- Scientific data is most useful if it is:

<i>self-describing</i>	for independent use
<i>portable</i>	for current and future platforms
<i>directly accessible</i>	for efficient access to subsets
<i>appendable</i>	for incremental creation
<i>sharable</i>	for concurrent access and writing
<i>archivable</i>	for future uses of past archives

- Preserving backward compatibility, for both APIs and format, is sacrosanct.
- Simplicity of the interface and generality for multiple disciplines are also desirable.

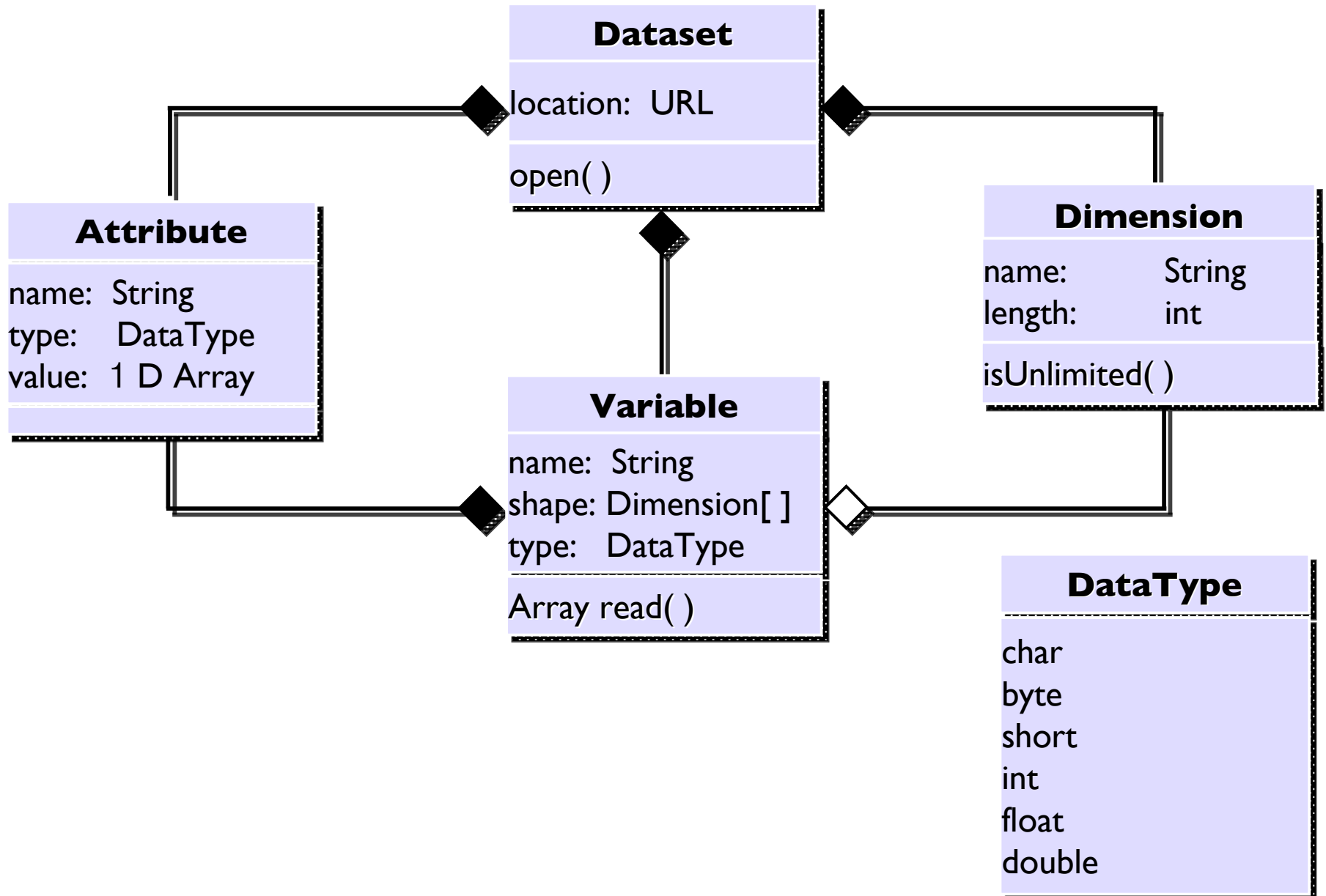
# What is netCDF-4?

- A NASA-funded joint project combining desirable characteristics of netCDF and HDF, while taking advantage of their separate strengths
  - Widespread use and simplicity of netCDF
  - Generality and performance of HDF5
- Improves interoperability with other scientific data representations, support for high-performance computing
- Currently in alpha release, first general release expected later this summer

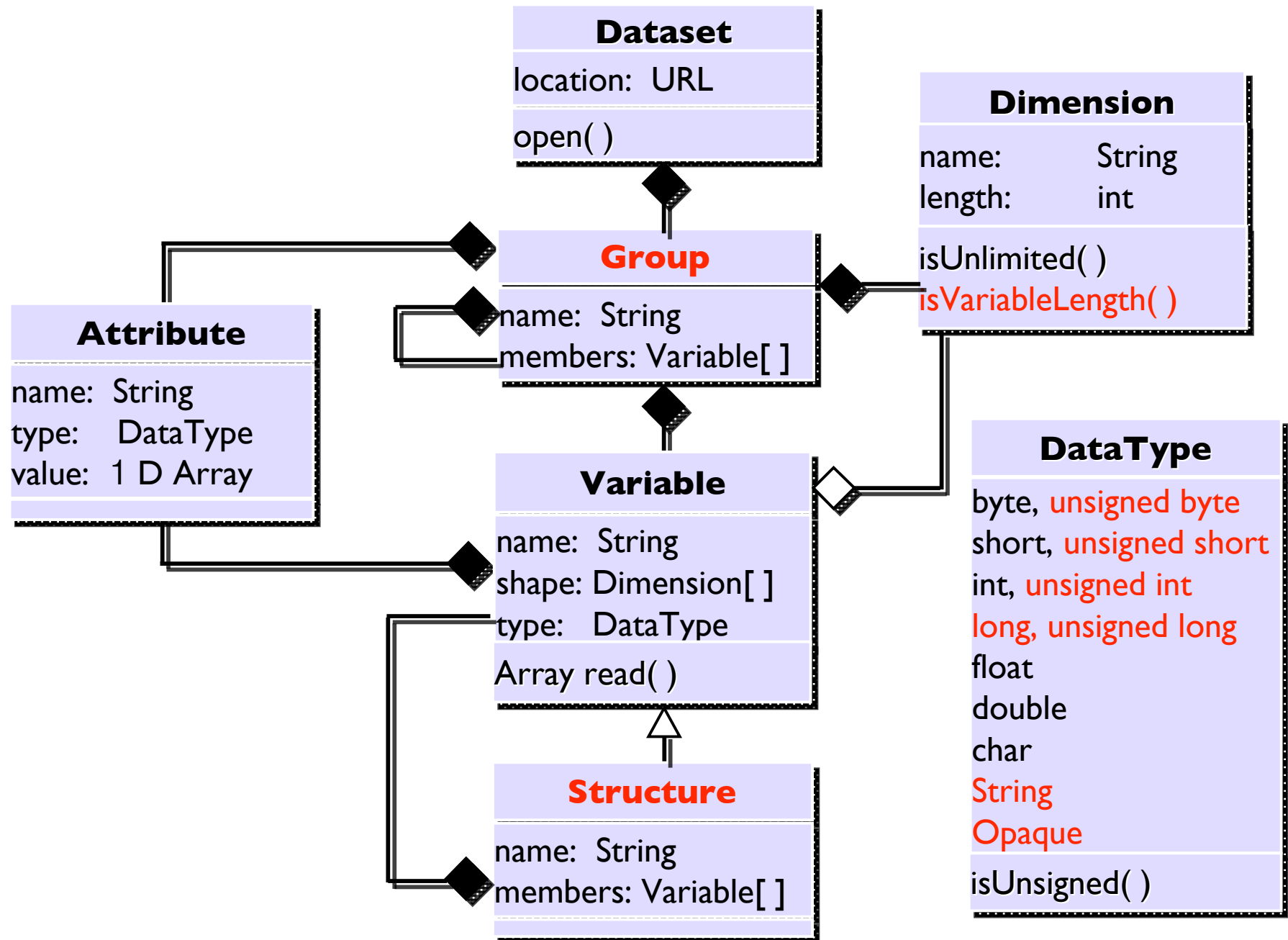
# NetCDF-3 and NetCDF-4 Data Models

- NetCDF-3 models multidimensional arrays of primitive types with Variables, Dimensions, and Attributes, with one unlimited dimension
- NetCDF-4 implements an extended data model with:
  - Structure types: like C structs
  - Multiple unlimited dimensions
  - Groups: containers providing hierarchical scopes for variables, dimensions, attributes, and other groups
  - Variable-length objects: for soundings, ragged arrays, ...
  - New primitive types: Strings, unsigned ints

# NetCDF-3 Data Model



# NetCDF-4 Data Model

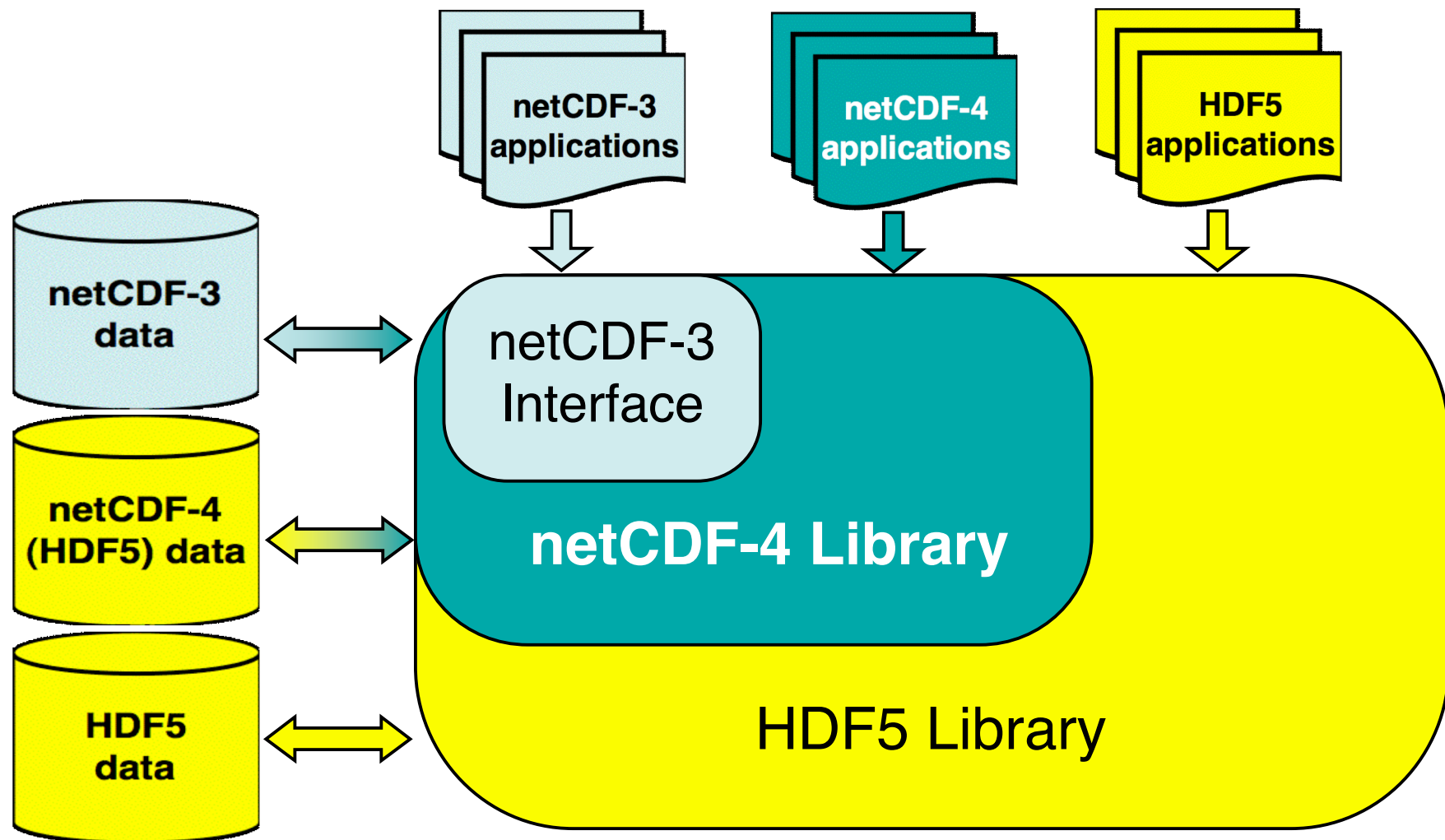




# A Common Data Model?

- NetCDF, HDF5, and OPeNDAP developers have discussed a mapping among the three data models
  - Opportunity to tweak the data models to mitigate differences
  - Opportunity to make OPeNDAP 4.0 the remote access protocol for netCDF-4 and netCDF-4 the persistence format for OPeNDAP
- This will take some time

# C Interfaces for netCDF and HDF5



Access to netCDF-3, netCDF-4, and HDF5 data  
created through netCDF-4 interface

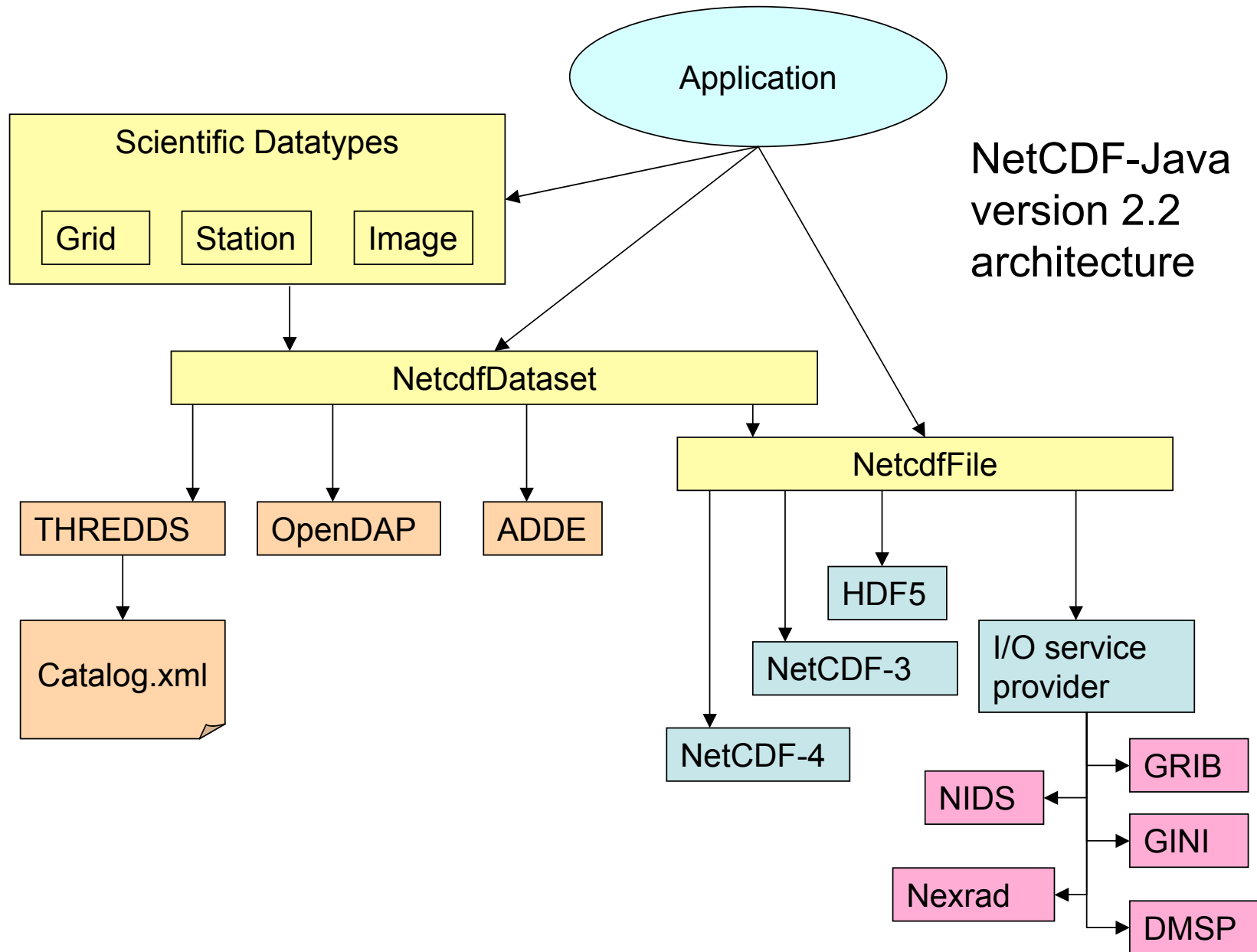
# How Are the APIs Changing?

- Current APIs for C, Fortran, Java, and C++ will continue to be supported
- NetCDF-4 features will initially be available only for C and Java interfaces, followed by Fortran-90 and eventually C++
- The Fortran-77 interface is frozen
  - Access from Fortran-77 to most netCDF-4 features is limited or not available (e.g. Structures)
- Advanced Java features will eventually be moved to C-based interfaces

# Advanced Features of Java Interface

- Supports client access to data servers:
  - HTTPD
  - OPeNDAP
- Supports access through NcML virtual datasets to add metadata, aggregate data, subset
- Java netCDF version 2.2 (in alpha release) implements
  - NetCDF-4 Data Model
  - Coordinate system support for general and georeferenced coordinates
  - I/O Framework providing netCDF interface to data in other formats: GRIB, HDF5, GINI, NEXRAD, ...

# NetCDF Java



# NetCDF-4 Format

- Still supports classic XDR-based format (1988) and 64-bit offset format variant (2004)
- Adds support for HDF5 representation to permit use of
  - Appending along multiple unlimited dimensions
  - Dynamic schema modification
  - Per-variable chunking (tiled storage)
  - Per-variable compression
  - Unicode names
  - “Reader makes right” conversions
- For maximum interoperability, stick to classic format

# Implications for Conventions

- Recommendation: delay using netCDF-4 features until best practices are clear
- Community conventions should be *very conservative* with respect to new versions of libraries and formats
- Structures ought to be useful for observational data, such as station data, soundings, trajectories, and profiles
- Groups may be useful for organizing complex datasets, ensembles, multiple sets of metadata conventions, nested meshes, ...

# Udunits Support

- During the next year, udunits will be included with netCDF
- Future netCDF development plans include modest udunits additions
  - logarithmic units such as dB
- Other possible enhancements depend on resources
  - XML syntax for units table
  - multiple units namespaces, for discipline-specific extensions or overrides



# Summary

- The current data model, APIs, and format will be supported into the indefinite future
- The netCDF-4 release adds structs, multiple unlimited dimensions, groups, new data types
- Will netCDF be made irrelevant by binary XML dialects?